THIRD SEMI-ANNUAL REPORT
ON
RESEARCH ON

CONTROL OF FREE-FLYING
SPACE ROBOT MANIPULATOR SYSTEMS

Submitted to
Dr. Henry Lum, Jr., Chief, Information Sciences Division
Ames Research Center, Moffet Field, CA 94035

by

The Stanford University Aerospace Robotics Laboratory
Department of Aeronautics and Astronautics
Stanford University, Stanford, CA 94305

Research Performed Under NASA Contract NCC 2-333
During the period March 1986 through August 1986

Principal Investigator:
Professor Robert H. Cannon Jr.

Introduction

This report reviews work performed during the period March through August 1986 on NASA Contract NCC 2-333 by the Stanford Aerospace Robotics laboratory for NASA's Ames Research Center. The goal of our research is to develop and test control strategies for self-contained, autonomous free-flying space robots. Such robots would perform operations in space similar to those currently handled by astronauts during extravehicular activity (EVA). Use of robots should reduce the expense and danger attending EVA both by providing assistance to astronauts and in many cases by eliminating altogether the need for human EVA, thus greatly enhancing the scope and flexibility of space assembly, maintenance, and repair activities.

The focus of our work is to develop and carry out a program of research with a series of physical Satellite Robot Simulator Vehicles (SRSV's), two-dimensionally freely mobile laboratory models of autonomous free-flying space robots such as might perform extravehicular functions associated with operation of a space station or repair of orbiting satellites. (It is planned, in a later phase, to extend our research to three dimensions by carrying out experiments in the Space Shuttle cargo bay.) Preceding progress under this contract is reported in References 1 and 2.

The SRSV facility we are developing (in room 010B of Stanford's Durand Bldg.) is described in Reference 2, where our first vehicle is shown on the smaller (6′ by 12′) of two granite tables the facility will have. The larger table (9′ by 12′ and weighing 16 tons) has been purchased from the quarry in North Carolina and is currently in San Jose where riggers are preparing it for the journey into our laboratory.

On these tables our space-robot vehicles can be operated (in planar motion) with a g level of about 10^{-5} and a friction-drag-to-weight ratio of about 10^{-4} — quite a good approximation to the situation in space.

We are currently pursuing three experimental projects with the SRSV facility: (1) Initial experiments with a robot vehicle using a single, two-link manipulator; (2) Development of a more advanced two-arm robot vehicle that will interact with other vehicles (representing small spacecraft or

other free-flying robots), and (3) Development of Systems for autonomous navigation and obstacle avoidance/target rendezvous by free-flying robots. The first experiments are led by Harold Alexander, under whose direction the SRSV facility has been built. The two-arm experiment development is being led by Ross Koningstein, and the guidance system work by Marc Ullman. All three people are Ph.D. Candidates and Research Assistants.

Reports on progress made during this report period on the three experimental projects are contained in the remainder of this document and the attached paper, entitled "Experiments on the Control of a Satellite Manipulator." In addition, a separate document is attached, a report by Marc Ullman entitled," 'Star Tracker' Surrogate Vision System for Mobile Robot Navigation"

In addition to the above experiments, on which progress is reported here, we have recently been urged to pursue new research in the area of autonomous control of free-flying robots. The request was made by Dr. Colladay, during his September (1986) visit to our laboratory, that we propose formally to increase our level of effort in order to pursue issues of space robot autonomy beginning immediately. If our proposal (Ref. 3) is accepted, Stanley schneider will lead this new effort, as he described to Dr. Colladay.

It should be noted in closing this introduction that our work on space robots described here continues to draw heavily on major efforts in other areas of our laboratory: fundamental flexible-link robot adaptive control research sponsored by AFOSR, development of two-cooperating-arm manipulator control systems for DARPA, and research in advanced manufacturing-robot concepts for our consortium of industrial sponsors.

Control of a Two-Arm Free-Flying Robot

Space assembly and manipulation will require robots to perform many complex tasks. As the complexity of tasks increases, so must the complexity of the robots and their associated control systems. Two-armed robots will be capable of performing many tasks not easily performed by single-armed robots. This section discusses some preliminary work toward research in the control of two-armed satellite robots.

Multiple cooperating arms will bring benefits in several areas of orbital robotic assembly and maintenance. Tasks that involve handling large objects will require the application of large torques to control their orientation. This is not feasible with single-armed robots due to the concentrated stresses that would occur at the manipulator/object interface. The application of large torques is much easier for two-armed robots because of the moment arm available due to the separated points of contact.

Mobility is an essential characteristic of any robot intended to operated in the orbital environment. Using thrusters is wasteful of fuel if it is possible to substitute a *climbing* and/or *jumping* strategy for travel about a large structure. In climbing, and especially, jumping, the robot will need to control both its position and attitude. Attitude control requires the application of large torques. Similarly, throwing and catching objects in space will require control of their position and attitude.

Control system complexity is the main difficulty encountered in rendering functional a robot with multiple arms. The primary requirement is cooperation and coordination between the two arms. Selectively compliant control, for example, allows positioning of a large object held between two grippers, without stretching or crushing the object held. Such undesirable forces are caused by imprecise control of the two arm positions, or positioning inconsistent with the robot's grasp on the object. Selective compliance allows a manipulator to be positioned along one (or more) axes, while a certain force characteristic is maintained along the remaining axes, notably along the imaginary line joining the two points of contact on the object being manipulated.

It is also necessary for an orbital robot to handle objects of arbitrary mass properties. For precise control of a body's trajectory, the robot will

need to know that body's mass, mass center, and moment of inertia. These characteristics may be learned by experimentation with the object once gripped, or they may need to be identified in real time as the object is being caught. Such adaptive control is being pursued on several fronts in our laboratory.

Many tasks will require moving and orienting payloads. This can only be accomplished in a microgravity environment by using vehicle thrusters. In order to achieve accurate control during periods of combined thrusting and manipulation, the arm controller must be able to compensate for large forces on the vehicle body imposed by attitude and position thruster operation.

These control problems will form the basis of our study. They will also influence the design of the two-armed satellite simulator vehicle.

Design of the Two-Armed Satellite Simulator Vehicle

The vehicle is to be built on a 50 cm-diameter pad, to be floated using the same air-bearing technology used in the single-arm Satellite Robot Simulator Model. This pad will support the compressed gas tanks that provide gas for flotation and thrusters. Also to be placed on the main body pad are the guidance and control computer and the actuators (motors) for the two arms.

The robot will be equipped with two manipulator arms which have a large intersecting workspace to maximize the area over which cooperation is possible. Actuation for the shoulder and elbow joints is to be done with limited angle torquers. Laboratory experience has shown that these actuators possess excellent torque linearity with no brush friction. Their main limitations are their angular range (typically ± 70 degrees) and a slightly lower electrical efficiency than brush commutated motors. The two arms are to be mounted spaced apart at the vehicle body so that large objects can be grasped. The end effectors will consist of compliant force-sensing wrists.

The robot will have thrusters located far away from the actuator arms so that thrust does not need to be diverted from the workspace (for example, using multiple thrusters at angles away from the workspace). A gas jet provides a significant disturbance to arm motion if the arm is in the path

of the jet. In order to make the most efficient use of thrust, the gas thrusters will be located in pods suspended out over the sides of the vehicle body. This way, the only disturbance to arm control introduced is through motion of the vehicle body.

Finally, the mobile satellite manipulator system will be spending its time between manipulation in a workspace, manipulation while in motion, or in transit between workstations. The transit requirement is severe because the robot must use internal energy sources to provide kinetic energy, whether through thrusters or by using its arms to propel it from one truss to another in some space structure. Consequently, the mass of the vehicle will be mainly in the power sources (ie. on-board electrical supply and pressurized gas for thrusters). The low mass of the two-link arms will introduce slight flexibility into arm motion at high bandwidth operation. However, our laboratory has demonstrated the ability to control manipulators significantly more flexible.

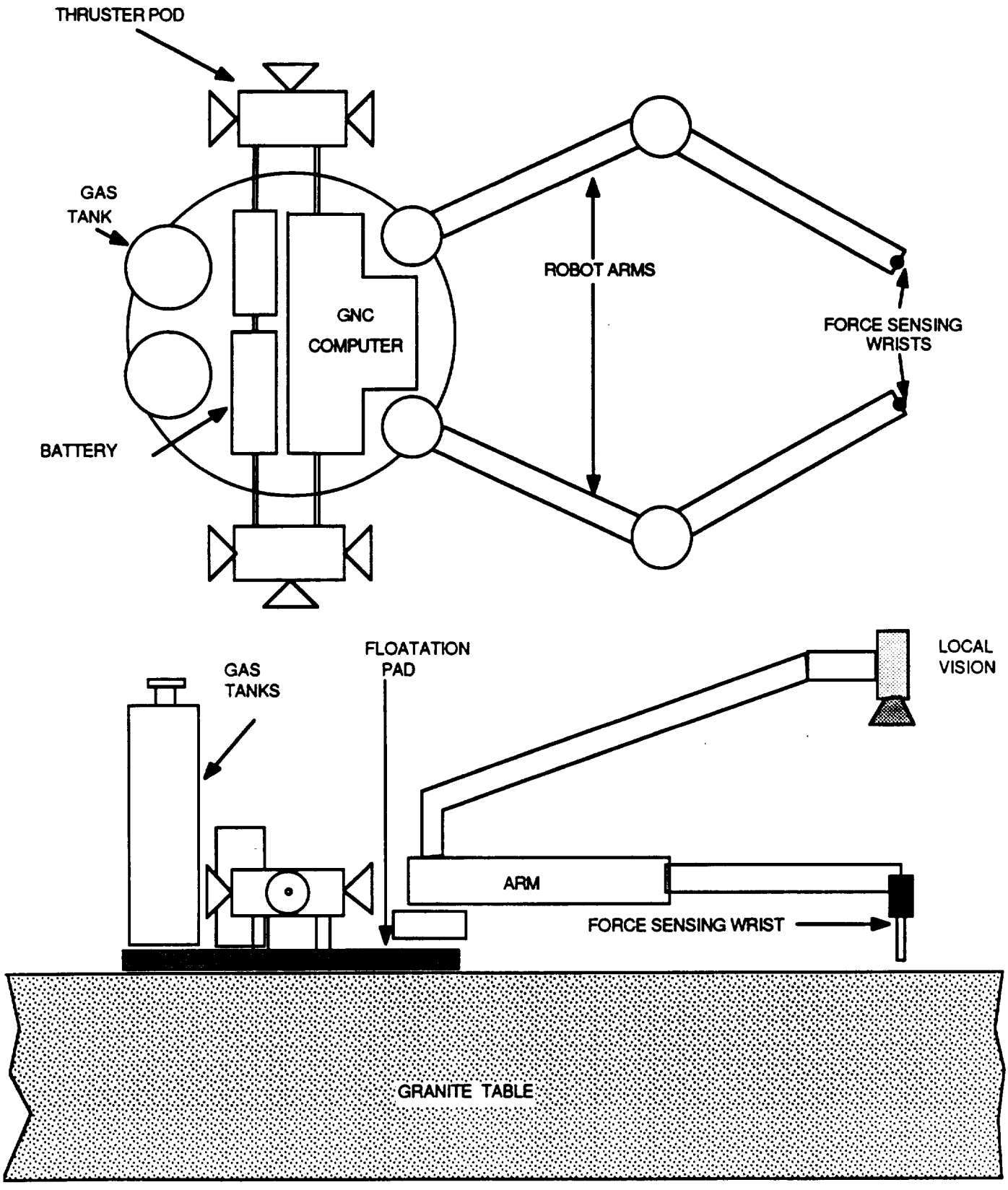
Figure 1 shows a simplified diagram of the two-armed satellite manipulator to be constructed. Scaling of the vehicle size was done considering the size of the table surface we have to work with (about 3×4 meters) and the desirability of placing more than one vehicle and target on the table together. In scaling the vehicle's (and arm's) mass and power requirements, the desire to be able to move the manipulator tip at roughly 1 *meter/second* after 1 second under maximum actuation was a design goal. This speed will not create an inordinate computational demand on the control computer selected, nor will it be so slow as to make manipulation a tedious process. Main limitations in laboratory vehicle optimizations turn out to be due to commercially available hardware for thruster gas supply storage, and the mass of electrical power sources. A desired vehicle on-line time between gas refills and/or electrical power pack replacement is on the order of five minutes (maximum thrusting) to an hour (exhaustion of electrical supply).

Demonstrations of System Capabilities

We propose a set of demonstrations of two-armed robot system capabilities. They are chosen to show the mastering of the control problems associated with two-armed control, and also to be readily identifiable as being applicable to real world (ie. orbital assembly) problems.

Figure 1

TWO ARM SATELLITE ROBOT SIMULATOR



5.A.

The first demonstration involves taking a two-dimensional body of arbitrary mass and determining its mass and inertia characteristics. Then it will be moved along a predefined trajectory. This will demonstrate cooperating arms control and system adaptability to various loads, and is applicable to general manipulation for assembly or transportation.

The long term goal of the demonstration is the two-armed throwing and catching of arbitrary two-dimensional rigid bodies, which will show real-time identification of object mass and inertia properties. This illustrates more advanced control and identification capability and is inherently applicable to space construction, satellite despinning, and a host of other tasks.

Conclusions

We have begun the research into a two-armed satellite simulator vehicle by identifying the areas of control to be mastered. These areas are adaptive control, selective compliance for arm cooperation, and thruster disturbance rejection. The vehicle design was carried out with the understanding that it is to be a testbed for new control designs aimed at furthering the knowledge in these fields.

Autonomous Navigation and Control of Free-Flying Space Robots

Work leading to the demonstration of autonomous control of an air cushion based Satellite Robot Simulator Vehicle (SRSV) has been progressing on several fronts. A hierarchy for overall system management has been devised and now serves as the backbone for control system organization. A first cut at the high level operator interface for task specification has been demonstrated. A preliminary architecture for the new SRSV has been developed which stresses modularity for ease of system construction and servicing. These areas are discussed in greater detail below.

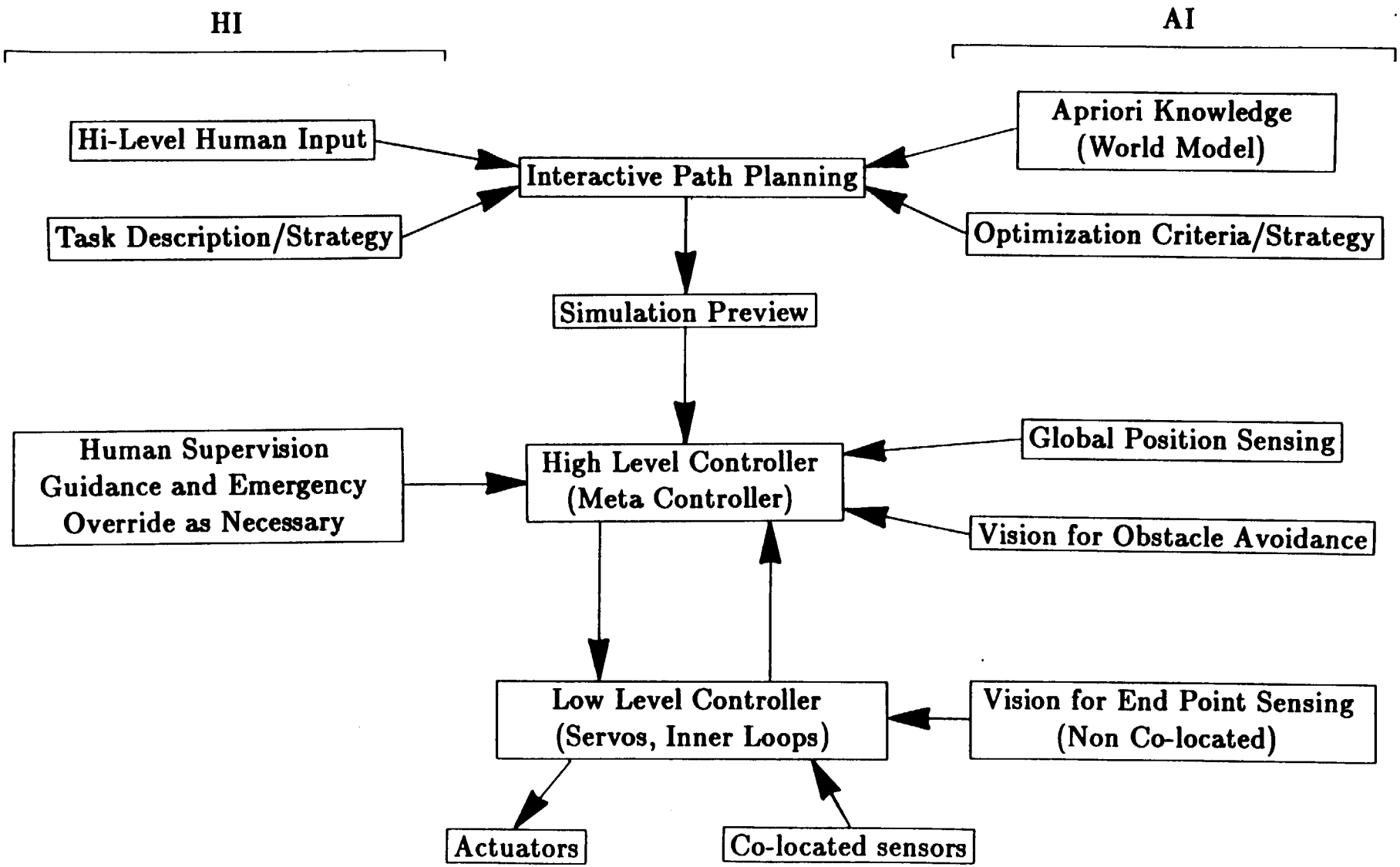
Hierarchy for Global / Strategic Control

In order to manage a highly autonomous system as complex as a free-flying robot one must have an overall organizational structure. The natural choice is one which is hierarchical in nature. We have chosen a scheme based on this premise which is depicted in Fig. 2. The human operator will enter task level command descriptions via an interactive graphical interface. These commands may be such things as starting and ending points, desired way points, and target locations. Using this information, as well as á priori knowledge of the vehicle configuration, dynamics, and capabilities, a computerized path generation system will display a proposed path. The operator may then elect to carry out the required task in simulation or accept it as being satisfactory in which case it will be executed by the actual system.

To perform the requested task, the generated path information is passed down the hierarchy to a meta-controller which oversees the operation of several lower-level controllers. This meta-controller will receive information from a global positioning system and a local vision system so as to be able to detect potential collisions with unforeseen obstacles. A real time path correction system running as one of the meta-controller sub-tasks will attempt to revise the pre-planned path to avoid any such collisions. Only if this system is unable to handle such an exception condition will it be

Figure 2

Hierarchy for Global / Strategic Control



necessary to notify the human operator to ask for assistance in deciding how to proceed.

The low level servo systems will maintain tight closed loop control of the vehicle base (using the thrust/propulsion system) and the arm configuration (via link actuators) using local sensors and global information. The end point control capabilities which have been developed previously by the ARL will be incorporated into this system. By controlling the endpoint directly rather than using joint angles and kinematics to predict the tip position, the floating base problem can be successfully mastered.

New Vehicle Design

In order to facilitate the extension of our work into global control and navigation, we have begun designing a new vehicle. A preliminary configuration has been selected pending the availability of certain critical components. The new design emphasizes modularity and space utilization so as to minimize the overall package size without compromising serviceability. The desired improvements in the new vehicle design also include:

- A smaller and lighter base plate consisting of an aluminum honeycomb sandwich construction.
- A more efficient propulsion system which combined with the smaller and lighter vehicle yields a better thrust to weight ratio for improved maneuverability.
- A more advanced on board computer system using state of the art 32 bit mircoprocessor technology networked to a series of off-board computers.
- This vehicle will operate on a new 9 x 12 foot granite surface plate thereby giving it a much larger area to operate in.

A vision based global positioning system has been prototyped that allows the vehicle to track its position as it moves around on the surface plate. A local vision system based on the same principles can be used for locating targets as well as the manipulator endpoint. This system offers the following features:

- It can operate over a theoretically infinite area.
- It can be made robust to occlusion problems.
- It can be adapted to track multiple targets.
- It can be made to operate fairly rapidly.

The drawbacks are:

- It needs to be complemented with some form of absolute measurement so as to alleviate drift problems.

A complete report on this system which details the underlying theory and initial experimental results is available under separate cover.

Operator Interface

A preliminary version of an interactive graphical operator interface has been developed which allows the operator to use a mouse or other similar pointing device to select weigh points and targets on a computer generated view of the "world." This view shows a planar representation of boundaries, pre-identified obstacles, and possible starting and ending points. After the operator has described the desired task by selecting various objects and/or points by "pointing and clicking," the computer generates a proposed trajectory and superimposes it on the graphical model. If the user accepts the proposed path, he can elect to view an animated simulation of the task to be executed. Once this simulation has been played out, the operator may elect to revise the path or desired goals, or he may accept it in which case it would then been carried out by the physical hardware (which does not yet exist).

Future Progress

Within the next six months, the new surface plate should be in place with the new vehicle built and operating (without its arm). (The arm will be added after control of the base has been achieved). Hopefully the global sensing system will be functioning so that fully autonomous trajectories can be executed.

References

[1] Cannon, R. H. Jr., "First Semi-Annual Report on Research on Control of Free-Flying Space Robot Manipulator Systems", Stanford University Aerospace Robotics Laboratory, Stanford, CA, September 1985.

[2] Cannon, R. H. Jr., "Second Semi-Annual Report on Research on Control of Free-Flying Space Robot Manipulator Systems", Stanford University Aerospace Robotics Laboratory, Stanford, CA, March 1986.

[3] Cannon, R. H. Jr., "Basic Research for Manipulation Strategies for Autonomous Space Robots: Plan to Increase Effort for Year Three of NASA Cooperative Agreement NCC 2-333.", Stanford University Aerospace Robotics Laboratory, Stanford, CA, October 1986.

Experiments on the Control of a Satellite Manipulator

Harold L. Alexander *
Robert H. Cannon, Jr. †
Aerospace Automation Laboratory
Department of Aeronautics and Astronautics
Stanford University
Stanford, California 94305

September 25, 1986

Abstract

Automation is becoming increasingly important to the exploration and utilization of space. Space-based robotic systems will provide efficient and inexpensive means to work in space. The dynamic control of space robots presents unique challenges, partly due to the robot's lack of a fixed base. This paper discusses a theoretical and experimental study of the control of satellite robots, focusing on the control of a two-dimensional laboratory model supported on air bearings, and equipped with a two-link robotic arm. The air bearings provide the same freedom of motion in the two dimensions of the supporting surface as orbiting bodies experience in three dimensions. The satellite model is a dynamic system with five degrees of freedom. Techniques are being tested that will be applicable to the case of a three-dimensional satellite robot possessing many more degrees of freedom. Control of the satellite model has been simulated, and control of its two-link arm has been demonstrated with the base fixed.

Introduction

Space robots will perform many tasks in space similar to those currently handled by astronauts during extravehicular activity (EVA). Use of robots will reduce the expense and danger attending EVA by helping astronauts or eliminating the need for human EVA for some tasks. If these operations are rendered safe and inexpensive, it will greatly enhance the scope and flexibility of space assembly and repair activities.

This report reviews experiments in the dynamic control of a single-arm space robot. Specifically, we are interested in the problem presented by the lack of a fixed base from which the robot can measure and control its manipulator position and by the tendency of its free-floating body section to respond to motions of the attached manipulator arm.

Our control research is focused on a laboratory model of such an orbital robot. Control of the robot is obtained via a modified form of resolved-acceleration control (Wampler, 1984; Luh et al., 1980; Khatib, 1987), using a mathematical model of the robot's dynamic response to arm actuation. This is similar to other lines of research being conducted by the Stanford University Aerospace Automation Laboratory, which concentrate on noncolocated end-point control of robots (Tilley and Cannon, 1986; Cannon and Schmitz, 1984; Rosenthal, 1984; Maples, 1985). This paper describes the techniques intended to control the satellite robot model, and preliminary experiments in the control of the robotic arm with fixed base.

*PhD Candidate, Electrical Engineering.
†Charles Lee Powell Professor and Chairman.

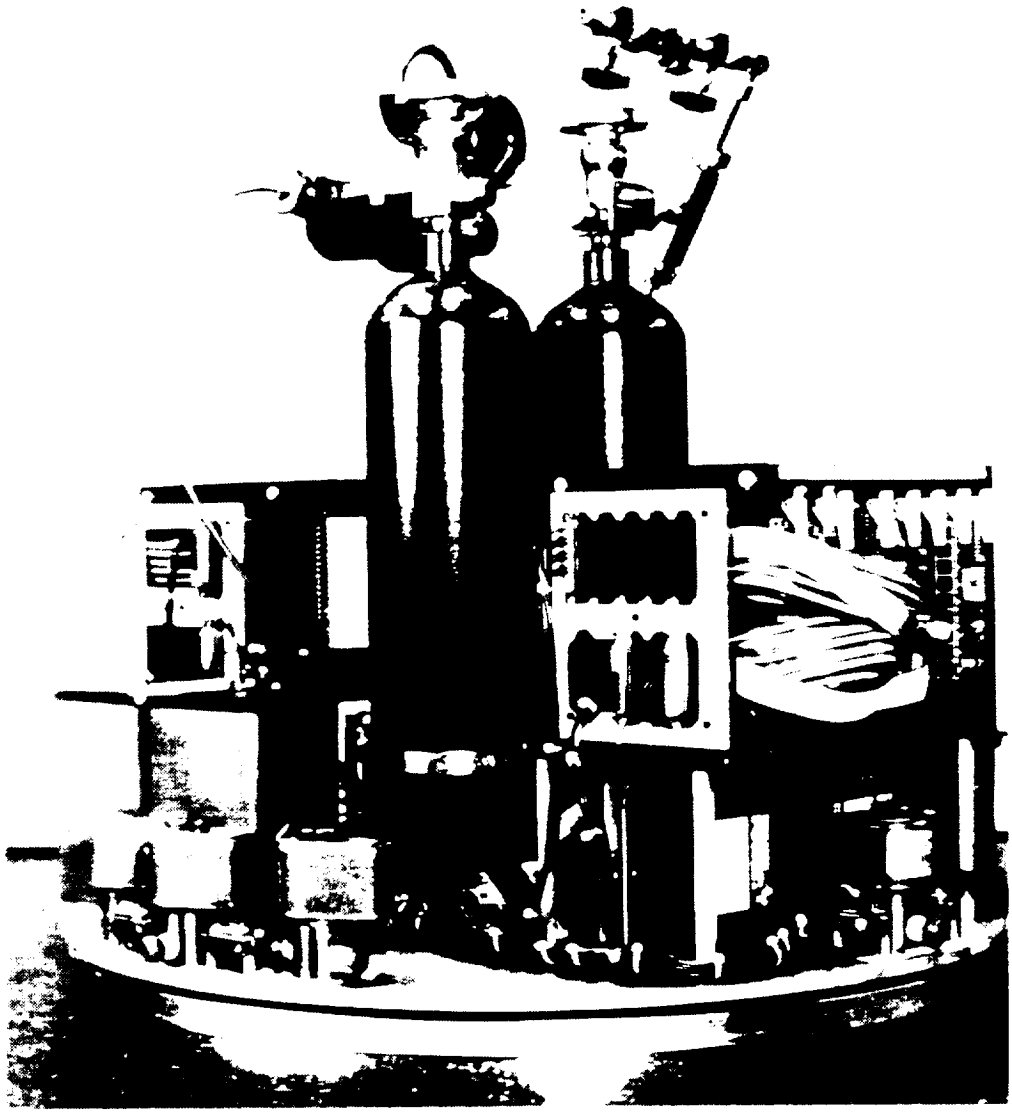


Figure 1: *The Satellite Robot Simulator Vehicle is shown in an early form without its arm. The computer is shown to the right; one of the two batteries is shown to the left behind a set of thrusters. The nitrogen tanks at center supply support and thruster gas.*

The Satellite Robot Simulator Vehicle

Figure 1 is an early photo of the Satellite Robot Simulator Vehicle (SRSV) without its arm. The main body of the robot is represented by a two-foot-diameter air-cushion vehicle, supported on a film of gas approximately .005 inches thick (Rehsteiner, 1968). The gas film is maintained between the base of the vehicle and a large granite surface plate (table) measuring 6×12 feet and ground flat to an accuracy of .001 inches. The base of the vehicle is also machined flat to .001 inches. The table is carefully leveled to eliminate gravity-induced accelerations of the vehicle. The vehicle is equipped with eight thrusters for computer control of attitude and position. An angular rate sensor is also included to sense rotation of the main body .

The SRSV is equipped with a single two-link arm. The arm joints are supported on individual air-cushion pads of their own and operate in the two dimensions of the granite table surface. Each joint of the arm is equipped with a direct-drive brushless torque motor for control and an optical encoder for angle sensing. A television camera locates targets that are marked with infrared light-emitting diodes.

The SRSV possesses two degrees of freedom in translation and one in rotation of the base body,

and the arm joints provide two additional degrees. While the system possesses three redundant degrees of freedom for control of the arm tip position, only the two arm joint actuators will generally be used for such control. The point of this restriction is that thruster gas is saved by avoiding frequent corrections to the base body's position. Thruster gas is considered to be a particularly scarce commodity in orbit. The model thus represents a two-input, two-output plant with three uncontrolled DOF's. It is allowing for these extra degrees of freedom that presents an unusual control challenge.

Measurement of Model Parameters

Accurate control generally depends on accurate dynamic and kinematic modeling of the plant. The inertial parameters of the SRSV were measured by weighing each part, and by observing the period of oscillation of a rotary pendulum loaded with each part in turn. They are presented below along with important dimensional parameters (see figure 2).

l_1	$.371m$	Distance from the center of the main body to the shoulder axis. The main body has its mass center at its geometric center.
l_2	$.254m$	Length of the upper arm, measured from the shoulder axis to the elbow axis.
l_3	$.254m$	Length of the forearm, measured from the elbow axis to end effector.
l_2^*	$.224m$	Length from shoulder to mass center of upper arm
l_3^*	$.236m$	Length from elbow to mass center of forearm.
M_1	$31.2kg$	Mass of base body.
M_2	$2.08kg$	Mass of upper arm
M_3	$1.02kg$	Mass of forearm
I_1	$1.47kg\text{-}m^2$	Moment of inertia of base body. Each such moment is about a vertical line passing through the respective mass center.
I_2	$.011kg\text{-}m^2$	Moment of inertia of upper arm.
I_3	$.0095kg\text{-}m^2$	Moment of inertia of forearm.

Dynamic Analysis of SRSV

A schematic diagram of the SRSV is shown in Figure 2. Its equations of motion have been derived using Kane's system of dynamic analysis (Kane, 1985). They are presented below in equations 1-29 in a matrix form.

The basic relationship between actuator forces and accelerations is expressed:

$$F(t) + F^*(q, u, \dot{u}) = 0 \tag{1}$$

Substituting $F' - M\dot{u} = F^*$, as explained below,

$$\begin{aligned} F(t) + F'(q, uu^T) &= M\dot{u} \\ &= MA \end{aligned} \tag{2}$$

where $F(t)$ is a vector of actuator forces,

$$F(t) = \begin{bmatrix} F_1(t) & F_2(t) & F_3(t) & F_4(t) & F_5(t) \end{bmatrix}^T \tag{3}$$

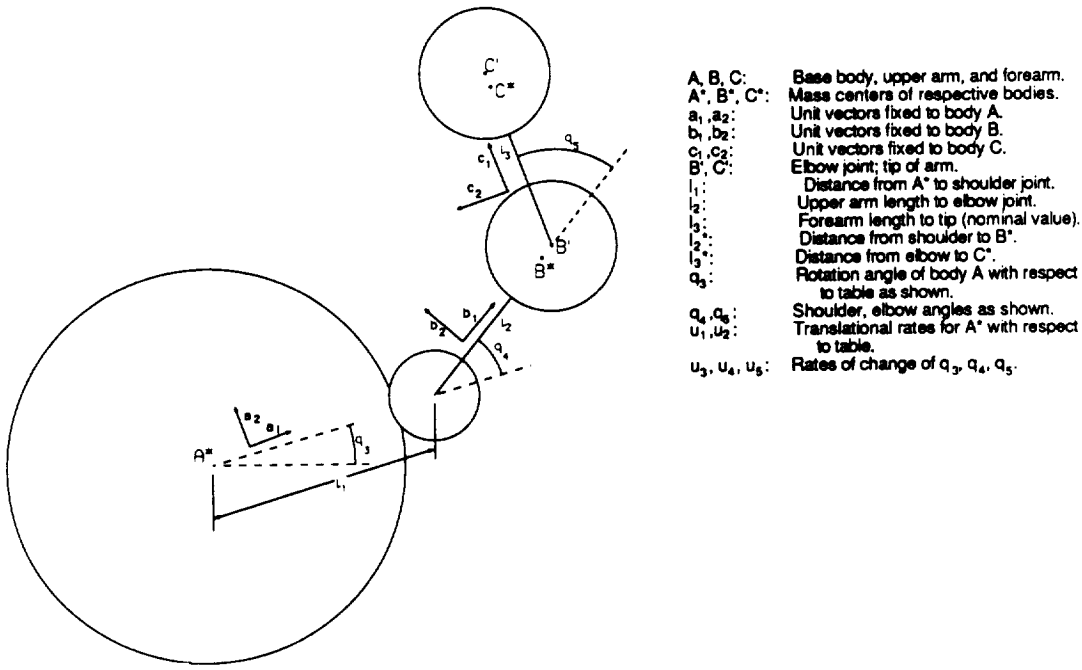


Figure 2: *SRSV Schematic Diagram. The base body of the vehicle and the arm pads are shown as disks; quantities used for analysis are shown to the right. The shoulder motor is extended from the base on a pylon to provide more freedom for the shoulder joint.*

q is a vector describing the configuration of the system (Figure 2),

$$q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 \end{bmatrix}^T \tag{4}$$

u is a vector of rates,

$$u = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 \end{bmatrix}^T \tag{5}$$

$F^*(q, u, \dot{u})$ is a vector of generalized inertia forces acting about each degree of freedom,

$$F^* = \begin{bmatrix} F_1^* & F_2^* & F_3^* & F_4^* & F_5^* \end{bmatrix}^T \tag{6}$$

and F^* is divided into two parts,

$$F^*(q, u, \dot{u}) = F'(q, uu^T) - M(q)\dot{u} \tag{7}$$

where F' is a vector of the nonlinear forces (coriolis and centripetal) acting about each degree of freedom,

$$F' = \begin{bmatrix} F_1' & F_2' & F_3' & F_4' & F_5' \end{bmatrix}^T \tag{8}$$

and the 5×5 mass matrix M contains all information about the involvement of the accelerations $\{\dot{u}_1, ..., \dot{u}_5\}$,

$$M = \begin{bmatrix} M_{11} & 0 & M_{13} & M_{14} & M_{15} \\ 0 & M_{22} & M_{23} & M_{24} & M_{25} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} \end{bmatrix} \tag{9}$$

$$M_{11} = M_{22} = M_{123} \tag{10}$$

$$M_{13} = M_{14} = -M_3 l_3^* s_{45} - (M_2 l_2^* + M_3 l_2) s_4 \tag{11}$$

$$M_{15} = -M_3 l_3^* s_{45} \tag{12}$$

$$M_{23} = M_3 l_3^* c_{45} + (M_2 l_2^* + M_3 l_2) c_4 + M_3 l_1 + M_2 l_1 \tag{13}$$

$$M_{24} = M_3 l_3^* c_{45} + (M_2 l_2^* + M_3 l_2) c_4 \tag{14}$$

$$M_{25} = M_3 l_3^* c_{45} \tag{15}$$

$$M_{33} = 2M_3 l_1 l_3^* c_{45} + 2M_3 l_2 l_3^* c_5 + 2(M_2 l_1 l_2^* + M_3 l_1 l_2) c_4 + I_3 + M_3 l_3^{*2} + M_3 l_2^2 + M_3 l_1^2 + I_2 + M_2 l_2^{*2} + M_2 l_1^2 + I_1 \tag{16}$$

$$M_{34} = M_3 l_1 l_3^* c_{45} + 2M_3 l_2 l_3^* c_5 + (M_2 l_1 l_2^* + M_3 l_1 l_2) c_4 + I_3 + M_3 l_3^{*2} + M_3 l_2^2 + I_2 + M_2 l_2^{*2} \tag{17}$$

$$M_{35} = M_3 l_1 l_3^* c_{45} + M_3 l_2 l_3^* c_5 + I_3 + M_3 l_3^{*2} \tag{18}$$

$$M_{44} = 2M_3 l_2 l_3^* c_5 + I_3 + M_3 l_3^{*2} + M_3 l_2^2 + I_2 + M_2 l_2^{*2} \tag{19}$$

$$M_{45} = M_3 l_2 l_3^* c_5 + I_3 + M_3 l_3^{*2} \tag{20}$$

$$M_{55} = I_3 + M_3 l_3^{*2} \tag{21}$$

c_i and s_i represent the sine and cosine of the ith coordinate q_i. Repeated subscripts are used to indicate summation, *e.g.* $u_{34} = u_3 + u_4$, $c_{34} = cos(q_3 + q_4)$, $M_{12} = M_1 + M_2$.

F', the vector of nonlinear forces, has the following elements:

$$F_1' = M_{123}u_2u_3 + M_3l_3^*c_{45}u_{345}^2 + (M_2l_2^* + M_3l_2)c_4u_{34}^2 + M_{23}l_1u_3^2 \quad (22)$$

$$F_2' = -M_{123}u_1u_3 + M_3l_3^*s_{45}u_{345}^2 + (M_2l_2^* + M_3l_2)s_4u_{34}^2 \quad (23)$$

$$\begin{aligned} F_3' = {} & M_3l_1l_3^*s_{45}(u_{345}^2 - u_3^2) - M_3l_3^*s_{45}u_2u_3 - (M_2l_2^* + M_3l_2)s_4u_2u_3 \\ & +(M_2l_1l_2^* + M_3l_1l_2)s_4(u_4^2 + 2u_3u_4) \\ & -M_3l_3^*c_{45}u_1u_3 - (M_2l_2^* + M_3l_2)c_4u_1u_3 \\ & -M_{23}l_1u_1u_3 + M_3l_2l_3^*s_5(u_5^2 + 2u_4u_5 + 2u_3u_5) \end{aligned} \quad (24)$$

$$\begin{aligned} F_4' = {} & -M_3l_3^*s_{45}u_2u_3 - M_3l_1l_3^*s_{45}u_3^2 - (M_2l_2^* + M_3l_2)s_4u_2u_3 \\ & -(M_2l_1l_2^* + M_3l_1l_2)s_4u_3^2 - M_3l_3^*c_{45}u_1u_3 \\ & -(M_2l_2^* + M_3l_2)c_4u_1u_3 \\ & +M_3l_2l_3^*s_5(u_5^2 + 2u_4u_5 + 2u_3u_5) \end{aligned} \quad (25)$$

$$F_5' = -M_3l_3^*s_{45}u_2u_3 - M_3l_1l_3^*s_{45}u_3^2 - M_3l_3^*c_{45}u_1u_3 - M_3l_2l_3^*s_5u_{34}^2 \quad (26)$$

Note that F' contains all terms which are nonlinear in $\{u_1, ..., u_5\}$. M describes the linear relationship between the active and nonlinear generalized forces $F + F'$, and the accelerations $\dot{u}$.

The velocity ${}^T\mathbf{v}^{C'}$ of the tip C' of the arm with respect to the table T may be expressed via the Jacobian J:

$${}^T\mathbf{v}^{C'} = u^T J \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} \quad (27)$$

$$J = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -l_2s_4 - l_3s_{45} & l_1 + l_2c_4 + l_3c_{45} \\ -l_2s_4 - l_3s_{45} & l_2c_4 + l_3c_{45} \\ -l_3s_{45} & l_3c_{45} \end{bmatrix} \quad (28)$$

The derivative of the Jacobian, $\dot{J}$, is used in the control algorithms:

$$\dot{J} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -l_2c_4u_4 - l_3c_{45}u_{45} & -l_2s_4u_4 - l_3s_{45}u_{45} \\ -l_2c_4u_4 - l_3c_{45}u_{45} & -l_2s_4u_4 - l_3s_{45}u_{45} \\ -l_3c_{45}u_{45} & -l_3s_{45}u_{45} \end{bmatrix} \quad (29)$$

Control Design

The SRSV and orbital robots present a unique problem in control design compared to ground-based robots. Lacking any rigid support, the body of the robot will respond to all motions of the attached arm. When the arm is extended to grasp an object, for example, the body will generally accelerate away from the target in response the arm's motion. The robot thus lacks any fixed reference from which its tip position and orientation may be determined and controlled. Any precise manipulator activity relative to an inertial target must allow for both the free relative motion of the target and robot and for the response of the robot's body to any commanded movement of the arm.

An additional controller may operate simultaneously with our arm controller, in order to control the SRSV's position adjacent to the target. The purpose of such a controller would not be to fix the position of the SRSV, but to keep the target within the operating envelope of the arm.

Its operation would likely produce intermittent, on-off control forces. The arm controller should accept information fed forward from such a companion controller, to allow for the effect of thruster operation on the arm tip motion.

The control method we have developed for dealing with this problem is related to a method for the control of fixed-base robots. The *computed-torque* controller calculates the precise torques necessary at each time step for the robot's joint angles to follow a prescribed time history (Craig, 1986). It maintains a mathematical model of the robot in order to calculate these quantities.

For a manipulator with n degrees of freedom, the computed-torque method involves deriving an n-input, n-output relationship between the applied control efforts (motor torques) and the resulting accelerations about each joint. We have already done this for the SRSV above; in order to achieve control of the five rates $\{u_1, ..., u_5\}$ we need only use the basic dynamic equation:

$$F = M\mathcal{A}_{des} - F' \tag{30}$$

where $\mathcal{A}_{des}$ represents the desired set of accelerations $\{\dot{u}_1, ..., \dot{u}_5\}$ to follow the desired trajectory in $\{q_1, ..., q_5\}$. The prescribed actuator forces $\{F_1, ..., F_5\}$ must be applied at thrusters and torque motors to achieve the trajectory.

The computed-torque method is useful when a trajectory in $\{u_1, ..., u_5\}$ may be calculated and followed that results in the desired tip position trajectory. In the case of the SRSV, however, such a trajectory may be very easy to find, but difficult to follow. This is because the thruster controls $\{F_1, F_2, F_3\}$ are not directed by this controller; only two independent quantities, not five, may be controlled via the available torque controls F_4 and F_5. Control of the arm only excites the free, unconstrained motion of the body of the vehicle.

The *resolved-acceleration* method offers a partial solution to this problem (Luh et al., 1980; Wampler, 1984). Whereas the basic dynamic equations for a manipulator relate joint rates to actuator forces and torques, the resolved-acceleration method uses a similar system of equations relating manipulator position and attitude to actuator inputs. The method applies to a robot with the same number of controls as it has degrees of freedom but does not provide for the case of the SRSV where three redundant degrees of freedom exist, and where only two controls are available for the five-degree-of-freedom system. What is needed is a relationship between the two *available* control inputs (the arm torques F_4 and F_5), and the two required controlled quantities (arm x and y motion relative to the table).

Derivation of this relationship is as follows. ${}^T\mathbf{a}^{C'}$ refers to the acceleration of the tip of the arm, C', in the table frame T.

$$F + F' = M\mathcal{A} \tag{31}$$

$${}^T\mathbf{v}^{C'} = u^T J \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix} \tag{32}$$

$$\begin{aligned} {}^T\mathbf{a}^{C'} &= \frac{d}{dt}\left({}^T\mathbf{v}^{C'}\right) \\ &= \dot{u}^T J \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} + u^T \dot{J} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} + u^T J \begin{bmatrix} u_3\mathbf{a}_2 \\ -u_3\mathbf{a}_1 \end{bmatrix} \\ &= \left\{ \dot{u}^T J + u^T \dot{J} + u^T J \begin{bmatrix} 0 & +u_3 \\ -u_3 & 0 \end{bmatrix} \right\} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} \end{aligned} \tag{33}$$

We wish to control this quantity, ${}^T\mathbf{a}^{C'}$, to obtain a "desired" value, ${}^T\mathbf{a}^{C'}_{des}$, which may be calculated according to some simple rule. The following rule prescribes a critically-damped ($\zeta = 1$) approach to a target P by the arm tip C', where P is fixed with respect to the table and the approach trajectory has a time constant ω of 1 second.

$$\begin{aligned} {}^T\mathbf{a}^{C'}_{des} &= -2\zeta\omega\, {}^T\mathbf{v}^{C'} - \omega^2 \mathbf{r}^{PC'} \\ &= -2\, {}^T\mathbf{v}^{C'} - \mathbf{r}^{PC'} \end{aligned} \tag{34}$$

$$ {}^{T}\mathbf{a}_{des}^{G'} = \left\{ \dot{u}^T J + u^T \dot{J} + u^T J \begin{bmatrix} 0 & +u_3 \\ -u_3 & 0 \end{bmatrix} \right\} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} \tag{35} $$

We can extract the $\mathbf{a}_i$ without approximation,

$$ \dot{u}^T J = \begin{bmatrix} {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_1 \\ {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_2 \end{bmatrix}^T - u^T \dot{J} - u^T J \begin{bmatrix} 0 & +u_3 \\ -u_3 & 0 \end{bmatrix} \tag{36} $$

$$ \dot{u} = \mathcal{A} = M^{-1} \left(F + F' \right) \tag{37} $$

$$ \left\{ M^{-1} \left(F + F' \right) \right\}^T J = \begin{bmatrix} {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_1 \\ {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_2 \end{bmatrix}^T - u^T \dot{J} - u^T J \begin{bmatrix} 0 & +u_3 \\ -u_3 & 0 \end{bmatrix} \tag{38} $$

$$ \left(F + F' \right)^T M^{-1} J = \begin{bmatrix} {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_1 \\ {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_2 \end{bmatrix}^T - u^T \dot{J} - u^T J \begin{bmatrix} 0 & +u_3 \\ -u_3 & 0 \end{bmatrix} \tag{39} $$

$$ \begin{aligned} F^T M^{-1} J = & \begin{bmatrix} {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_1 \\ {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_2 \end{bmatrix}^T - F'^T M^{-1} J - u^T \dot{J} \\ & - u^T J \begin{bmatrix} 0 & +u_3 \\ -u_3 & 0 \end{bmatrix} \end{aligned} \tag{40} $$

Now partition F, the vector of actuator forces and torques, into two parts F_K and $F_{des}^{\dagger}$, known and desired. If an independent controller operates to maintain the position of the SRSV relative to the target, the thruster forces $\{F_1, F_2, F_3\}$ exerted by this controller may be passed to the arm controller as part of the vector F_K of known forces. The desired forces $F\dagger_{des}$ are those it is desired to apply to the arm joint motors.

$$ F_K = \begin{bmatrix} F_1 & F_2 & F_3 & 0 & 0 \end{bmatrix}^T \tag{41} $$

$$ F_{des}^{\dagger} = \begin{bmatrix} 0 & 0 & 0 & F_4 & F_5 \end{bmatrix}^T \tag{42} $$

$$ \begin{aligned} \left(F_K + F_{des}^{\dagger} \right)^T M^{-1} J = & \begin{bmatrix} {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_1 \\ {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_2 \end{bmatrix}^T - F'^T M^{-1} J - u^T \dot{J} \\ & - u^T J \begin{bmatrix} 0 & +u_3 \\ -u_3 & 0 \end{bmatrix} \end{aligned} \tag{43} $$

$$ \begin{aligned} F_{des}^{\dagger T} M^{-1} J = & \begin{bmatrix} {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_1 \\ {}^{T}\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_2 \end{bmatrix}^T - \left(F_K + F' \right)^T M^{-1} J \\ & - u^T \dot{J} - u^T J \begin{bmatrix} 0 & +u_3 \\ -u_3 & 0 \end{bmatrix} \end{aligned} \tag{44} $$

Define F_{des}, containing the nonzero elements of $F_{des}^{\dagger}$:

$$ F_{des} = \begin{bmatrix} F_4 \\ F_5 \end{bmatrix} \tag{45} $$

Define M_{des}^{-1} to include those elements of M^{-1} which are multiplied by nonzero elements of F_{des}' in equation 44:

$$ M_{des}^{-1} = \begin{bmatrix} \left(M^{-1}\right)_{1,4} & \left(M^{-1}\right)_{1,5} \\ \left(M^{-1}\right)_{2,4} & \left(M^{-1}\right)_{2,5} \\ \left(M^{-1}\right)_{3,4} & \left(M^{-1}\right)_{3,5} \\ \left(M^{-1}\right)_{4,4} & \left(M^{-1}\right)_{4,5} \\ \left(M^{-1}\right)_{5,4} & \left(M^{-1}\right)_{5,5} \end{bmatrix}^T \tag{46} $$

$$F_{des}^T M_{des}^{-1} J = \begin{bmatrix} {}^T\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_1 \\ {}^T\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_2 \end{bmatrix}^T - \left(F_K + F'\right)^T M^{-1} J$$
$$-u^T \dot{J} - u^T J \begin{bmatrix} 0 & +u_3 \\ -u_3 & 0 \end{bmatrix} \quad (47)$$

$$\left(M_{des}^{-1} J\right)^T F_{des} = \begin{bmatrix} {}^T\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_1 \\ {}^T\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_2 \end{bmatrix} - J^T M^{-1} \left(F_K + F'\right)$$
$$-\dot{J}^T u - \begin{bmatrix} 0 & -u_3 \\ +u_3 & 0 \end{bmatrix} J^T u \quad (48)$$

The matrix product $M_{des}^{-1} J$ is a 2×2 matrix, and easily invertible.

$$F_{des} = \left(M_{des}^{-1} J\right)^{-T} \left\{ \begin{bmatrix} {}^T\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_1 \\ {}^T\mathbf{a}_{des}^{G'} \cdot \mathbf{a}_2 \end{bmatrix} - J^T M^{-1} \left(F_K + F'\right) \right.$$
$$\left. -\dot{J}^T u - \begin{bmatrix} 0 & -u_3 \\ +u_3 & 0 \end{bmatrix} J^T u \right\} \quad (49)$$

F_{des} contains the torques to be exerted at the SRSV arm joints in order to achieve the acceleration prescribed by the rule above for $\mathbf{a}_{des}$. Thus the feedback rule for trajectory following has been established.

Simulation and Experimental Results

Figure 3 shows computer simulation results for a full-order controller for the floating-base condition of the SRSV. The interaction between arm and base is clearly seen: the vehicle possesses no initial angular velocity, but rotates back and forth as the arm swings to reach and hold its target. Some disturbance of the vehicle's translational motion to the right can also be seen as a up-and-down curvature of its path.

Figure 4 shows experimental results for the the arm controller that has been tested. A step input was used to challenge the algorithms's performance; normally a trajectory in position, velocity, and acceleration would be prescribed to improve tracking. Note that the path of the arm between destination points is a nearly straight line. This is a result of the formula for ${}^T\mathbf{a}_{des}^{G'}$:

$${}^T\mathbf{a}_{des}^{G'} = -2\,{}^T\mathbf{v}^{G'} - \mathbf{r}^{PG'} \quad (50)$$

For ${}^T\mathbf{v}^{G'} = 0$ at the beginning of a motion, tip motion will generally be in a straight line. Nonzero initial tip velocity will generally result in a curved path to the destination. The current rule provides no integral control, which we intend to add later in order to improve disturbance rejection.

Computed-torque and related methods are suspected of being strongly sensitive to imprecision in the specification of inertial parameters (Slotine, 1984). We have actually found experimental results for this simplified case that demonstrate robustness in the face of large variations of the tip mass.

Conclusions

A controller has been formulated for the control of a two-link arm mounted on a two-dimensional satellite model using air cushion support. Arm motions are prescribed with respect to inertial space or to a target. A simplified version of the controller, which does not recognize the freedom of the base to move, has been implemented. The full controller recognizing the free motion of the base is being implemented on the experimental SRSV. This controller is capable of allowing for disturbing control forces applied to the body of the model.

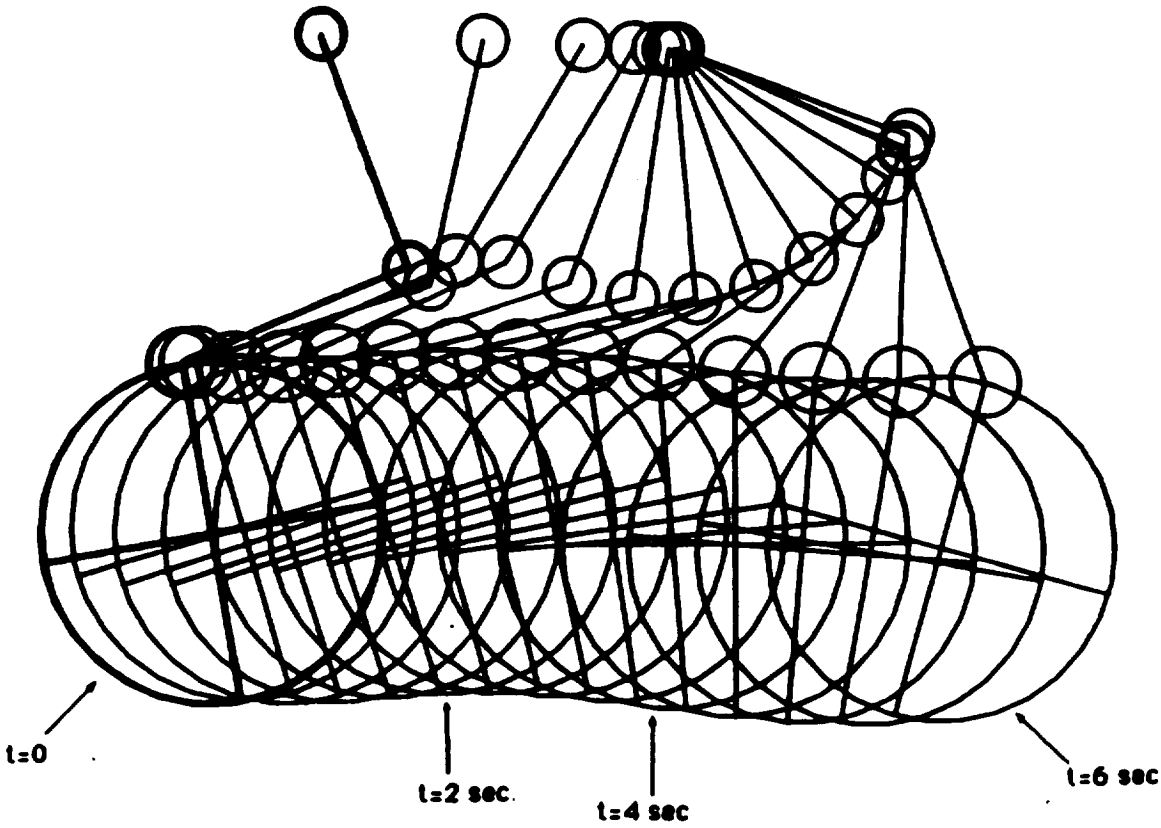


Figure 3: *Simulation results for the full 5-degree-of-freedom SRSV arm controller. The arm has been commanded to approach and hold at a target, from initial conditions which have the vehicle translating to the right at .25 m/sec. The specified response is as in Figure 3.*

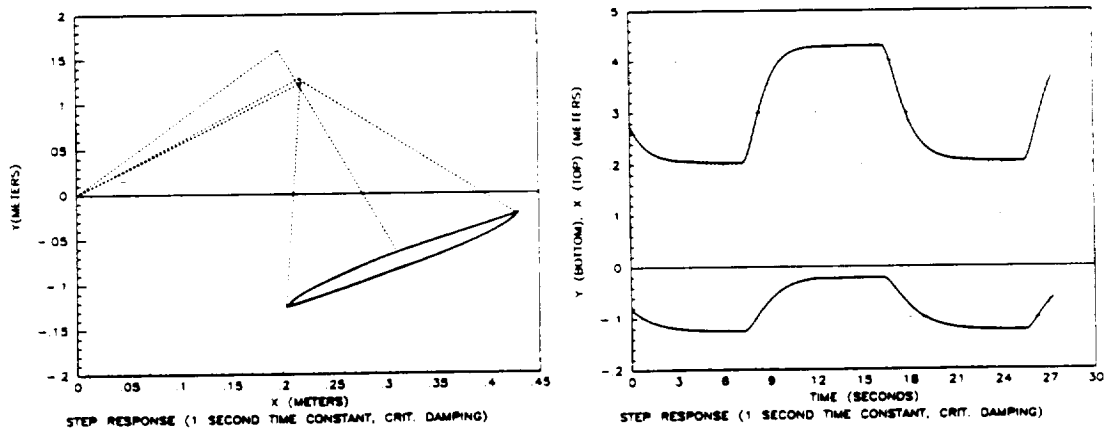


Figure 4: *Experimental results for a resolved-acceleration controller for the two-degree-of-freedom SRSV arm. The command is a step position change; the specified response has a second-order character with 1-second time constant and critical damping in each of x and y. Deviation from a straight line is ascribed to sampling rate and parameter errors.*

References

Rosenthal, D., 1984, "Experiments in Control of Flexible Structures with Noncolocated Sensors and Actuators," Ph.D. Thesis, Stanford University, Stanford, California.

Cannon, R. H. and Schmitz, E., 1984, "Initial experiments on the End-Point Control of a One Link Flexible Experimental Manipulator," *International Journal of Robotics Research*, The MIT Press, Cambridge, MA, Vol. 3, No. 3, pp. 62-75.

Craig, J. J., 1986, *Introduction to Robotics Mechanics & Control*, Addison-Wesley, Reading, Massachusetts.

Kane, Thomas R. and Levinson, David A., *Dynamics: Theory and Applications*, McGraw-Hill, New York, New York, 1985.

Khatib, O., 1987, "A Unified Approach for Motion and Force Control of Robot Manipulators: the Operational Space Formulation," *IEEE Journal of Robotics and Automation*, Vol. 3, No. 1.

Luh, J. Y. S., Walker, M. W., and Paul, R. P. C., 1980, "Resolved-Acceleration Control of Mechanical Manipulators," *IEEE transactions on Automatic Control*, Vol. AC-25, No. 3, pp. 468-474.

Maples, J., 1985, "End-Point Force Control of a Flexible Manipulator Arm", Ph.D. Thesis, Department of Electrical Engineering, Stanford University, Stanford, California.

Rehsteiner, F. H., 1968, "Static and Dynamic Properties of Hyudrostatic Thrust Gas Bearings with Curved Surfaces", Ph.D. Thesis, Department of Aeronautics and Astronautics, Stanford University, Stanford, California.

Slotine, J. J. E., 1984, "Robustness Issues in the Control of High-Performance Robots," Proceedings of the 23rd Conference on Decision and Control, Las Vegas, NV, December 1984.

Tilley, S. W. and Cannon, R. A., 1986, "End-Point Force Control of a Very Flexible Manipulator with a Fast End Effector," Proceedings of the Winter Annual Meeting, American Society of Mechanical Engineers, Anaheim, California.

Wampler, C. W., 1984, "Computer Methods in Manipulator Kinematics, Dynamics, and Control: a Comparative Study", Ph. D. Thesis, Department of Mechanical Engineering, Stanford University, Stanford, California.

"STAR TRACKER" SURROGATE VISION SYSTEM FOR MOBILE ROBOT NAVIGATION

Marc Ullman
Aerospace Robotics Laboratory
Stanford University

June 5, 1986

TABLE OF CONTENTS

TABLES AND FIGURES

I. INTRODUCTION

The Stanford University Aerospace Robotics Laboratory under the direction of Prof. Robert H. Cannon, Jr. has a NASA contract to design and build Satellite Manipulator Models for studying the use of autonomous robots in space. Possible applications for such space robots include the assembly and operation of the proposed space station as well as the repair of crippled satellites, a task which must currently be carried out by human astronauts. In order for these mobile robots to achieve autonomy, they must be able to navigate by themselves.

Laboratory simulation is currently in the form of an Air Cushion Vehicle (ACV) floating on a large granite table which allows nearly drag free dynamics to be studied on earth (in two dimensions). In order to focus attention on the navigation, obstacle avoidance, and fine position and attitude control problems a second table is being obtained having dimensions 9ft x 12ft. This new table will serve as the operating base for a new vehicle which will be substantially smaller and more deft than its predecessors. In order for the ACV to be able to navigate its way around this relatively large area it needs an accurate source for obtaining real-time position and orientation information. Since the system must be mechanically passive (i.e. no wires, etc.) so as to not affect the dynamics of the vehicle, it has been proposed to use some form of vision. This report describes the initial development and implementation of such a real time vision system for use in navigation and control of the ACV.

II. PREVIOUS WORK

Since one of the guiding principles behind the work done in the Aerospace Robotics Laboratory/Robotics Controls Group is the use of endpoint sensing for feedback control, there has already been a fair amount of investigation into schemes for tracking points. The major drawback of these approaches, which have usually been based on centroid position sensors, has been their limited working area--typically about 1m x 1m. Beyond this range they suffer accuracy and non-linearity problems. They are also all typically limited to tracking single objects unless complex time-sliced multiplexing techniques are employed.

An alternate approach is to use a frame-based sensor such as a TV camera. Previous work in this area has lead to the existence of a CCD camera and prototype "point grabber" interface board. This interface board scans an image frame for any pixels that exceed a predefined threshold. The horizontal and vertical coordinates of those pixels that do are recorded in a table which can be read by a computer through an I/O port.

Previous implementations made use of a fixed base camera looking down on the scene of interest; however, considerations of the required area of coverage show that such a scheme will also run into problems of limited resolution and the need for sophisticated optics in order to obtain a sufficiently wide viewing angle.

Hence it is proposed herein to invert this geometry and mount the camera on the vehicle so that it looks upward at a stationary pattern above.

III. PROPOSED METHODOLOGY

By inverting the geometry as described above we allow for the possibility of two distinctly different types of position sensing. Namely, absolute position sensing and relative or incremental position sensing. These two approaches have their respective advantages and disadvantages.

Absolute position sensing provides absolute position in global or world coordinates, but at the cost of large computation requirements because the basic scene recognition problem must be solved in order to determine location.

Relative position approaches require only that the present scene be correlated to the previous scene which is a much simpler operation. They do not suffer from partial occlusion problems and they require much less computation. Their main drawback as with all "dead reckoning" schemes (in which position is obtained by performing a discrete integration of velocity) is that any biased errors will cause the system to drift over time. Such errors in orientation can be particularly severe because they serve as multipliers when converting relative motion to absolute position.

Nonetheless, these two approaches can be combined in such a way as to exploit their respective advantages and reduce their corresponding drawbacks. The incremental scheme can be used to obtain the rapid information that is necessary for a high bandwidth real time closed loop controller while the absolute technique can be executed periodically to obtain a position fix and hence reset any accumulated drift errors. Although both schemes are therefore important, this paper will concentrate on the incremental approach because the immediate need is for real time information. The slower absolute position technique can be integrated at a later date.

IV. DESIGN SPECIFICATIONS

Before a formal methodology can be proposed it is crucial to have a clear and accurate description of the requirements and specifications that such a system must meet. The following specifications, while admittedly somewhat more stringent than actually required, serve as an ideal against which any achieved performance can be measured.

1. Utilize the existing CCD camera and "point grabber" hardware interface.

2. Develop a sufficiently simplified/optimized algorithm so that the system can run in real time with an update rate of 30 Hz with existing computational capabilities.

3. Produce X-Y position accuracy to within 1 mm and orientation to within 1 degree given a maximum vehicle velocity of 0.5 m/sec and a

maximum vehicle rotation rate of 1/4 rev/sec. This sub-pixel accuracy should be achievable through averaging and interpolation.

4. Provide the ability to work over a very large area (possibly many times the size of the area that the camera can view).

5. Ensure robustness to occlusion conditions such as at the edge of the table where there may not be a full field of LED's.

6. Include the capability for the system to be self initializing and self calibrating. i.e. when turned on, the vehicle/vision system should be able to find "home" without operator intervention.

V. SYSTEM DESCRIPTION

The proposed system consists of the following major components (See Figure 1):

1. Infrared LED's (stars)

   A stationary array of infrared LED's mounted on the ceiling approximately 2.5m above the ACV table. Various configurations are possible including rectangular or polar arrays with nominal spacings of 10 to 20 cm between adjacent LED's. The incremental algorithms described below are not sensitive to the LED geometry and hence the addition of an absolute position sensing scheme will probably have the greatest impact on the final geometry chosen.

2. CCD Camera

   An upward pointing CCD camera mounted on a vertical boom at the center of the ACV base with a lens system such that the viewing area on the LED array is roughly 1m in diameter. The lens is covered with an Infrared passband filter so as to block out all sources of visible light.

3. "Point Grabber" Interface Board / Interrupt Subroutine

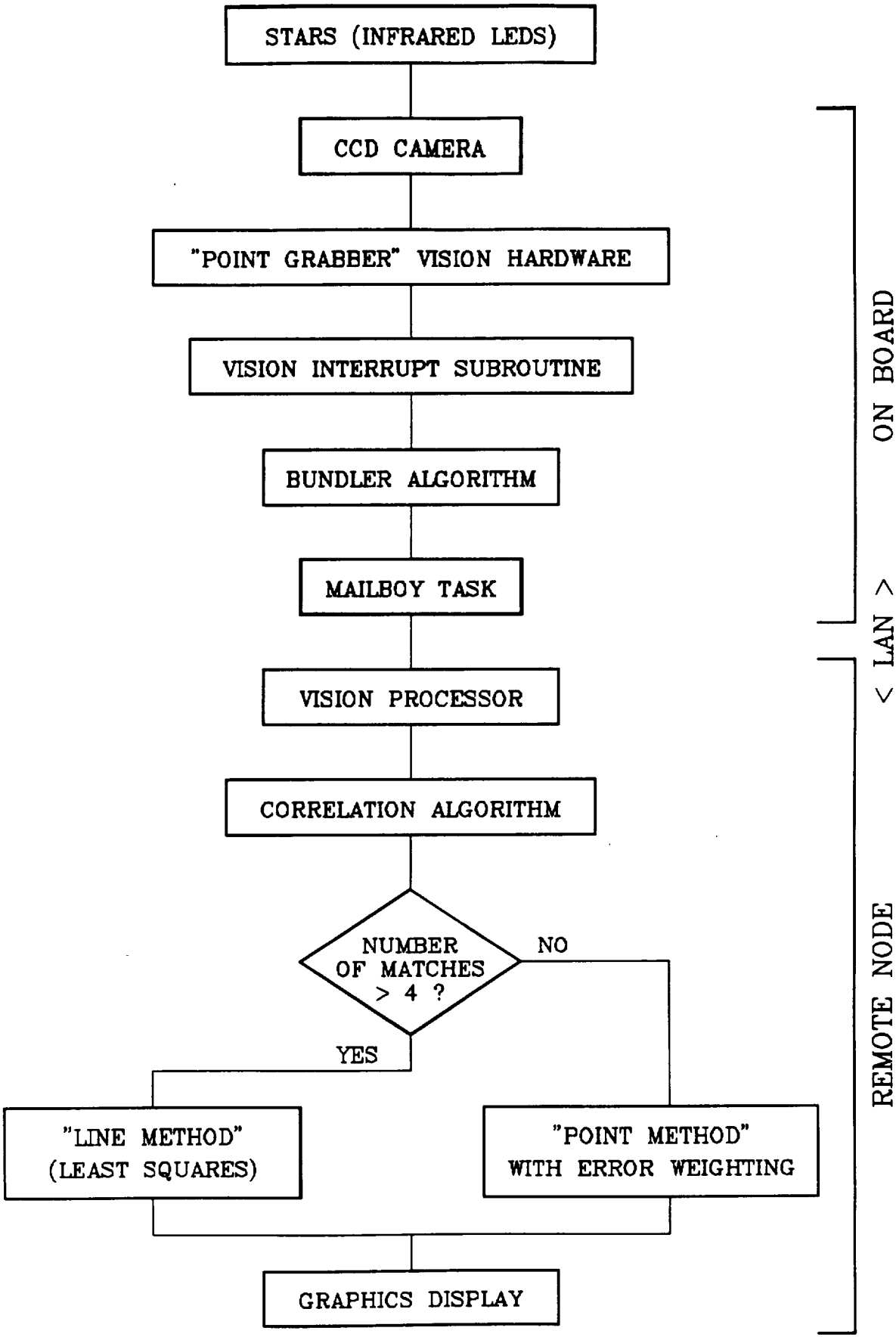
   A "point grabber" camera/computer interface device which scans each frame for pixels exceeding some predefined threshold and makes available a list of X-Y coordinates with respect to the camera frame. Upon completion of a scan consisting of both halves of an interlaced frame, the board generates a hardware interrupt which invokes an interrupt service routine in the onboard computer to read in the new data through an I/O port.

4. Pixel to Point Bundler Algorithm

   Due to blurring and blooming problems, a single LED can show up as several pixels. Hence the onboard computer system requires a software algorithm for bundling the pixel lists provided by the "point grabber" into a corresponding set of points. Associated with each

Figure 1

SYSTEM BLOCK DIAGRAM



point is a (x,y) coordinate pair indicating the estimated center of brightness of each LED.

5. Mailboy Data Passing Task

The computer architecture consists of a series of processors (nodes) connected via an ethernet LAN under the control of the QNX Network Operating System. Therefore, the actual vision processing can be accomplished off-board. A small task running on the onboard computer (referred to as the Mailboy) takes data from the bundler task and sends it out over the network to the vision processing node.

6. Vision Processor

The vision processor receives the list of (x,y) coordinates for each frame from the Mailboy task. It correlates the points in each new frame to those of the previous frame though a double variable length linked list sorting technique. The sorted data is then passed to an Inverse Kinematics type of algorithm which has been designed to handle the redundancy issue in such a way as to obtain a least squares or averaged solution. This use of redundant data reduces the noise sensitivity and improves the overall accuracy. The correlation algorithm and the position determination are the most substantial theoretical contributions of this work and are detailed below

7. Correlation Algorithm

The correlation algorithm must correct for the fact that two successive lists of point coordinates will most likely not correspond to the same set of physical points--one of the assumptions that the following Inverse Kinematics algorithms are based on. This lack of uniform order can result from a variety of causes. The two most common causes are:

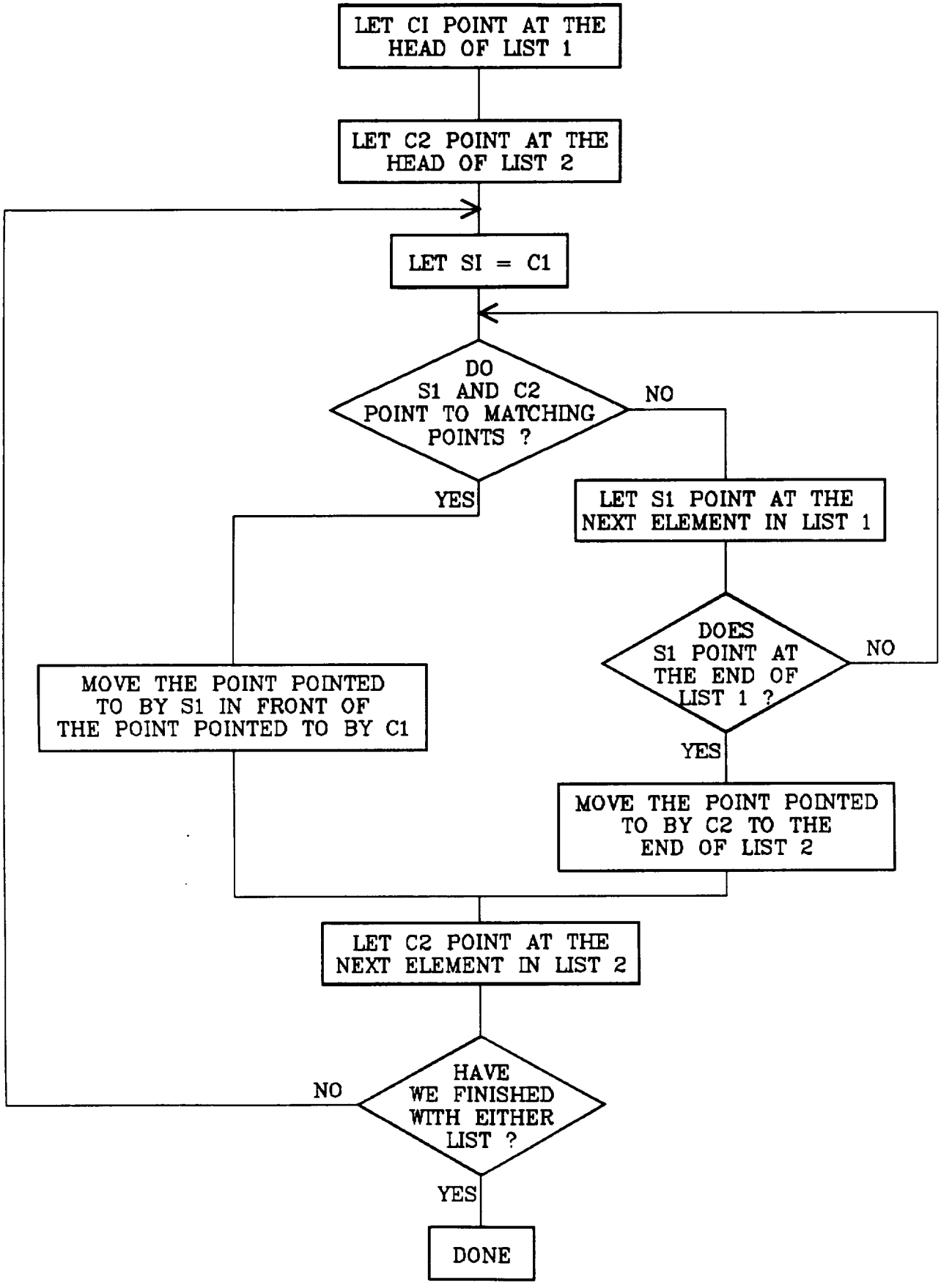
a. A point moving up or down one vertical scan line between frames so that it is switched to the opposite interlace thus leading to a large variation in its position within the list since all the points on one interlace are read in before those on the other.

b. Some points in the current frame may exit the field of view of the following frame while others may just be coming in to view. These unmatched points do not correlate at all and must be ignored for the purposes of kinematics. [The points that are just entering view for the first time are kept to see if they match points in the following frame even if they are not used in the current update.]

This sorting problem is solved by dumping the two lists of points into linked lists (See Figure 2). A set of travelling pointers then starts at the tops of the two lists. The first list is scanned for a match with the first entry in the second list. If a match is found, that entry is moved to the top of the first list and all other entries are pushed down. The two pointers then advance down their respective lists by one position.

Figure 2

BLOCK DIAGRAM OF CORRELATION ALGORITHM



If no match is found the entry on the top of the second list is moved to the bottom of that list. This process is then repeated until the shorter of the two lists has been scanned completely. The resulting lists contain all the matching entries at the tops of the lists followed by those points which have no matches.

8. "Inverse Kinematics" Algorithm

This algorithm takes the respective motions of each of the matched points and computes the corresponding rigid body rotation and translation that the camera frame must have undergone between the two scans. Two different algorithms were developed hence forth referred to as the "Line Method" and the "Point Method." The complete derivations and equations for both methods can be found in Appendix A.

The displacement vector corresponding to each point provides us with two each equations in three unknowns, the unknowns being t_x, t_y, and $d\theta$ -- the x and y translations and the rigid body rotation of the camera frame (See Figure 3). Therefore it takes a minimum of two points to determine a solution for the camera motion. Having more than two points provides us with more equations than unknowns and hence the problem becomes over constrained. We can use these extra constraints to obtain a least squares or averaged solution so as to reduce the overall noise sensitivity.

Since our design specifications limit the angular velocity of the camera frame to 1/4 rev/sec and our sample rate is 30 Hz, the maximum allowable value for $d\theta$ is ~0.05 rad. Hence, we are clearly justified in linearizing our equations with respect to $d\theta$. If we then eliminate $d\theta$ between each pair of equations, we end up with a set of conic equations representing the locus of possible translations inferred by each point. The problem then becomes one of finding the intersection of n circles (where n is the number of matched points) (See Figure 4).

The two algorithms mentioned above result from the fact that we can solve this problem in two ways. The "Line Method" works by finding the lines which pass though the two intersection points corresponding to each possible pairing of two circles. The intersection of these n(n-1)/2 lines can then be found using a pseudo inverse to obtain a least squares solution. Once we have t_x and t_y we can substitute them back into our full nonlinear equations to obtain estimates for $d\theta$, the corresponding rotation. At this point it might be asked, "Why not obtain $d\theta$ from the linearized equations?" Simulation results show that the actual determination of $d\theta$ is considerably more sensitive than that of t_x and t_y. Also, since the integral of $d\theta$, i.e. θ, is used to transform motion in the camera frame to position in world coordinates, any error in θ will cause the position updates to occur in improper directions. This sensitivity justifies solving the nonlinear equations for θ which requires performing two four-quadrant arc tangents.

Figure 3

TRANSFORM GEOMETRY

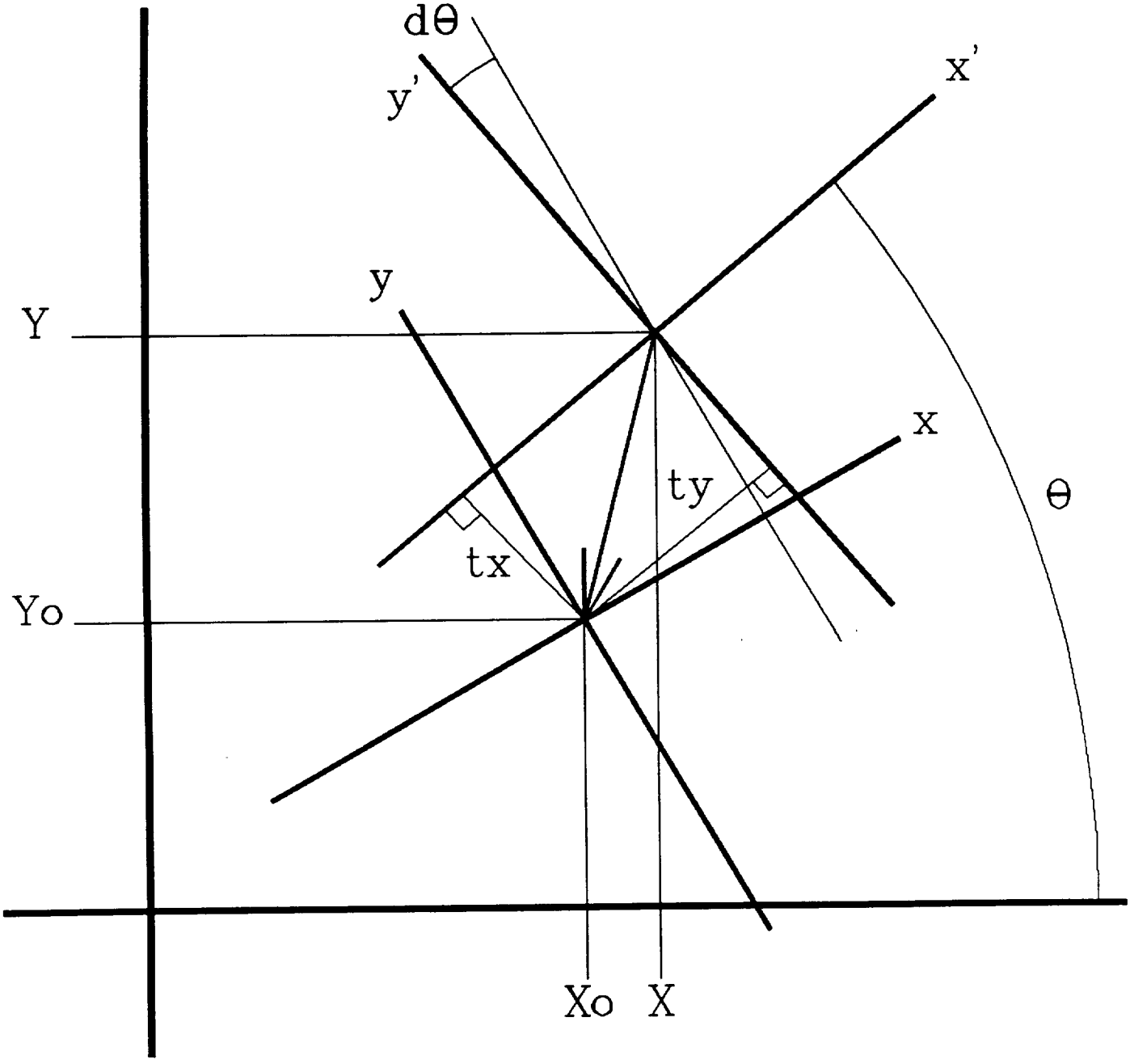
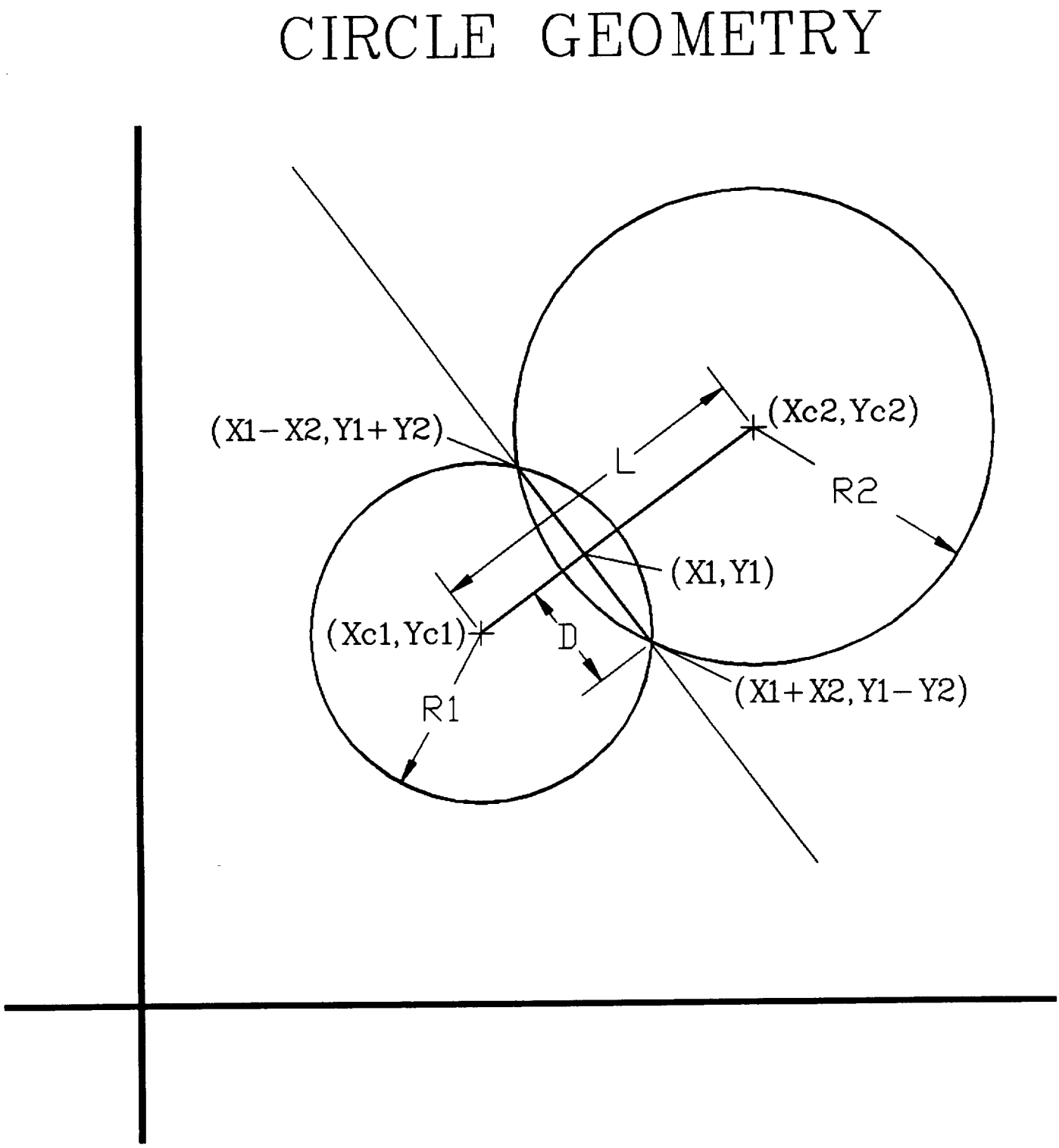


Figure 4


CIRCLE GEOMETRY
(X1−X2, Y1+Y2)
(Xc2, Yc2)
L
R2
(X1, Y1)
(Xc1, Yc1)
D
R1
(X1+X2, Y1−Y2)

The "Point Method" differs in that it makes use of the analytic solution for the intersection of two circles. This typically leads to two solutions and hence a consistency condition on the rotation $d\theta$ must be used to determine which is the correct solution. Again we obtain estimates for t_x and t_y and must turn to the nonlinear equations to solve for $d\theta$ which we have nearly already done to determine which solutions were consistent (See Appendix A).

The "Line Method" requires less computation because it involves solving only linear equations while the "Point Method" requires solving second order equations. The advantage of the "Point Method" is that it extracts more information from the data and hence yields more accurate results with fewer point pairs. Specifically the "Point Method" will always yield a solution for two or more points while the "Line Method" requires a minimum of three points and suffers from a singularity when all three points are collinear.

Both algorithms have been simulated using PC MATLAB and native C code implementations. Table 1 shows the relative accuracies of the two algorithms for various numbers of points using random camera motions and random point distributions. It is particularly interesting that the "Line Method" yields *better* accuracy than the "Point Method" as the number of points increases beyond n = 4. This result is due to the fact that the pseudo inverse provides a better solution than does simple averaging. The table also shows the results for a weighted average. The weighting is based on the reciprocal of the *angle consistency error* which is used to discriminate between the two possible solutions provided by the "Point Method." See Appendix A for a more detailed description of this quantity, but suffice it to say that is it a self consistency measure for each of the n(n-1)/2 solutions. Table 1 shows that this weighting improves the performance of the "Point Method" but the "Line Method" still yields the best performance for n > 4.

Table 2 gives the relative times required for computation of each algorithm. We see that the "Point Method" takes from two to five times longer than the "Line Method" because its requirements tend to increase as $n^2/2$ as opposed to n. Thus based on these results, the optimal solution is to use a hybrid combination of the two algorithms as shown in Figure 1. Since the "Point Method" with weighting can yield a valid solution for as few as two points and has reasonable computation requirements for n < 5, we use it for these cases so as to eliminate the problems of the "Line Method." For n > 4, the "Line Method" yields better accuracy at a much lower computational cost. Thus we can obtain the "best of both worlds" by using both algorithms.

9. Display Graphics

A set of graphics routines is used to plot the real time position (and eventually orientation) of the ACV/camera on a high resolution graphics terminal.

TABLE 1

STATISTICAL ERROR CHARACTERISTICS

(Normalized standard deviation of distance error
for random displacements and random star patterns)

Algorithm	N=2	N=3	N=4	N=6	N=9
Line	N/A	0.2302	0.0345	**0.0226**	**0.0114**
Point	0.1056	0.0409	0.0389	0.0353	0.0185
Point w/wtg	**0.1056**	**0.0293**	**0.0246**	0.0248	0.0142

TABLE 2

EXECUTION TIME REQUIREMENTS*

(times are in milliseconds)

Algorithm	N=2	N=3	N=4	N=6	N=9
Line	N/A	4.7	6.9	**12.6**	**24.5**
Point	4.2	10.1	18.8	45.0	105.8
Point w/wtg	**4.2**	**10.6**	**19.5**	46.9	109.5

* Timings are based on C implementation using Microsoft C Version 3.0 with inline 80287 double precision floating point math (See Appendix X for source code listing). They were run on a 12 MHz 1 wait state 80286 with a 14 MHz 80287 with 0 wait state interprocessor communications. Approximate system performance (for sake of comparison) is 1.5 mips CPU and 60 kflops FPU.

VI. EXPERIMENTAL RESULTS

The system described above has been tested in prototype form. Nine GE 1155 IR LEDs were spaced on 8 inch centers in a 3 x 3 rectangular array on a piece of peg board mounted on the laboratory ceiling. A Pulnix CCD camera with an IR passband filter was mounted on a tripod and aimed upward at the LEDs. The "Point Grabber" vision board was placed in an IBM PC-XT running at 8 MHz to simulate the "On Board" computer. The XT was networked via a Local Area Network running under the QNX multiprocessor/multitasking operating system to an IBM PC-AT running at 10 MHz with a 10 MHz 80287 Numeric Coprocessor. All of the realtime code was written in C using inline double precision floating point arithmetic except for the vision board interrupt subroutine which was coded in Assembly language. Graphical output was sent to a Selanar HiRez Graphics Terminal operating in Tektronix 4014 emulation mode.

Because it is simpler, the "Line Method" was the first algorithm to be tested in hardware. After extensive debugging (including the discovery that an updated version of the operating system was required) the system became operational. Although the code is unpolished and hence not running at the desired rate of 30 Hz, the initial results show extremely good position accuracy (about 1.5 mm) and very good stability (the noise appears to be unbiased). The performance does confirm the notion that the system is much more sensitive to orientation errors than to position errors. Problems result when the camera experiences fast or jerky motion but this type of motion will not be exhibited by the ACV due to its large mass/inertia to thrust ratio. The associated video tape shows the type of performance that has thus far been achieved.

VII. CONCLUSION

A new approach to using vision for obtaining real time position and orientation information for control of the ACV based Satellite Manipulator Model has been developed. Two algorithms have been derived and tested in computer simulations which demonstrate their respective advantages and disadvantages. The functionality of this new approach has been demonstrated by a hardware implementation under the QNX networked operating system as a multitasking/multiprocessor system. The initial results look quite promising and are encouraging enough to justify development of a full scale system. The only major problem seems to result from the fact that the code (in particular, the correlation algorithm) has not been optimized to run at 30 Hz so the effective system bandwidth is currently lower than desired.

VIII. AREAS FOR FURTHER RESEARCH

The following tasks will lead to a greater understanding of the underlying theory as well as to improved performance from the vision/tracking system.

1. Speed up and optimize the real time code so as to obtain a faster update rate. Possible improvements include switching to integer arithmetic, passing pointers instead of values to minimize data movement, and eliminating unnecessary debugging code.

2. Implement the "point algorithm" in hardware with and without error weighting to see if the theoretical advantages are realized in practice.

3. Filter the output and attempt to obtain velocity information as well as position.

4. Add an absolute position mode for finding home and for eliminating drift errors.

5. If a method can be obtained for discriminating between sets of LED's (or other light sources) (such as color or intensity) this approach could be used to simultaneously track multiple objects. Such a system would be very useful as an onboard vision system for viewing targets and/or obstacles that approach the ACV.

6. Although not necessary for the current application, the techniques proposed in this paper could be extended for use in three dimensional cartesian space (with constant orientation) by placing the LED's at a known spacing. The z-axis component (distance of the camera from the lights) could then be determined using perspective laws.

IX. REFERENCES

Ballard, Dana H. and Christopher M. Brown, Computer Vision, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, (C) 1982, pp 19.

Craig, John J., Introduction to Robotics, Mechanics and Control, Addison-Wesley Publishing Co., Reading, Massachusetts, (C) 1986, pp 109.

Kernighan, Brian W. and Dennis M. Ritchie, The C Programming Language, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, (C) 1978.

Microsoft C User's Guide and Reference Manual, Microsoft Corp., Bellevue, Washington, (C) 1985.

Moler, Cleve et al, PC-MATLAB User's Guide, The Mathworks, Inc., Sherborn, Massachusetts, (C) 1985.

Mortenson, Michael E., Geometric Modeling, John Wiley & Sons, New York, New York, (C) 1985, pp 345 - 355.

Papoulis, Athanasios, Probability, Random Variables, and Stochastic Processes, McGraw-Hill Book Co., New York, New York, (C) 1984, pp 108.

QNX C Compiler User's Manual, Quantum Software Systems Ltd., Ottawa, Canada, (C) 1986.

QNX Operating System User's Manual, Quantum Software Systems Ltd., Ottawa, Canada, (C) 1986.

Strang, Gilbert, Linear Algebra and Its Applications, Second Edition, Academic Press, New York, New York, (C) 1980, pp 103 - 150.

APPENDIX A - VISION/TRACKING ALGORITHM

Consider two successive frames from the camera, namely 1 and 2 respectively which each contain n uniquely identifiable points. Assume for the present time that each point in frame 1 corresponds uniquely to one point in frame 2 as follows:

$$(x_{1i},y_{1i}) \text{ <==> } (x_{2i},y_{2i})$$

If we assume that between these two frames the vehicle experienced a translation (t_x,t_y) and a rotation $d\theta$ then we can express the idealized relation between the i^{th} set of corresponding points as:

$$x_{2i} = (x_{1i} - t_x) \cos d\theta - (y_{1i} - t_y) \sin d\theta$$
$$y_{2i} = (x_{1i} - t_x) \sin d\theta + (y_{1i} - t_y) \cos d\theta$$

Since our problem statement includes the restriction that the maximum rotational rate shall not exceed 1/4 rev/sec we observe that with an update rate of 30 Hz, the maximum value for $d\theta$ is $2pi/30 \sim 0.05$ rad thus it is reasonable to linearize these equations with respect to $d\theta$ using the usual assumptions that:

$$\cos d\theta = 1 \text{ and } \sin d\theta = d\theta$$

With these substitutions our equations simplify to

$$(x_{1i} - t_x) - (y_{1i} - t_y)d\theta = x_{2i}$$
$$(x_{1i} - t_x)d\theta + (y_{1i} - t_y) = y_{2i}$$

Eliminating $d\theta$ between these two equations yields

$$(x_{1i} - t_x)^2 + (y_{1i} - t_y)^2 = (x_{1i} - t_x)x_{2i} + (y_{1i} - t_y)y_{2i}$$

By rearranging terms and "completing the squares" we can rewrite this equation in the familiar form of a circle centered at $(x_{1i} - x_{2i}/2 , y_{1i} - y_{2i}/2)$ with radius $r = [x_{2i}^2 + y_{2i}^2]^{1/2}/2$, namely:

$$[t_x - (x_{1i} - x_{2i}/2)]^2 + [t_y - (y_{1i} - y_{2i}/2)]^2 = (x_{2i}^2 + y_{2i}^2)/4$$

Thus for a collection of n points we have n circles which all intersect at a common point (t_x,t_y), or considering the effects of noise, in a common region. We can find this intersection point in one of two ways. The first method is to solve for the least squares intersection of a set of $n(n-1)/2$ lines while the second is to find the arithmetic mean of a set of $n(n-1)/2$ intersection points. The line method requires simpler computations but makes use of less information than the point algorithm.

THE LINE METHOD

In order to find the intersection points between each pair of circles, we observe that the intersection of any two circles, say those corresponding to the i^{th} and j^{th} points, is contained in the line passing through their zero, one, or two points of intersection. [In the case of zero, i.e. when the circles do not intersect, the resulting line lies between the circles and is perpendicular to the line joining their centers.] We can find this line by simply subtracting the equations of the two corresponding circles. To simplify the resulting equation we rewrite the above equation using the following notation:

$$(t_x - x_{ci})^2 + (t_y - y_{ci})^2 = r_i^2$$

Then the equation of the line passing through the i^{th} and j^{th} circles is:

$$2(x_{ci} - x_{cj})t_x + 2(y_{ci} - y_{cj})t_y = r_j^2 - r_i^2 + x_{ci}^2 - x_{cj}^2 + y_{ci}^2 - y_{cj}^2$$

Clearly this is a linear equation in our two unknowns t_x and t_y. If we have n intersecting circles, we will have $n(n-1)/2$ intersecting lines. If we rewrite the last equation as:

$$a_{k1}t_x + a_{k2}t_y = b_k \qquad \{ k: 1 \,..\, n(n-1)/2 \}$$

then we have a matrix formulation:

$$\mathbf{A}\,\mathbf{t} = \mathbf{b} \text{ where } \mathbf{t} = [t_x \; t_y]^T$$

which leads very naturally to a least-squares solution using the Moore-Penrose pseudo inverse, namely:

$$\mathbf{t} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

where $\mathbf{A}^T\mathbf{A}$ is a 2 x 2 matrix.

Now that we know t_x and t_y we can apply the well known solution for obtaining $d\Theta$ from our original nonlinear transformation equations, namely for each point pair $d\Theta_i$ is given by:

$$d\Theta_i = \text{ATAN2}(\, y_{2i}, x_{2i}\,) - \text{ATAN2}(\, y_{1i} - t_y, x_{1i} - t_x\,)$$

To reduce or noise sensitivity we simply average these values over the range i equals 1 to N.

THE POINT METHOD

We can also determine an explicit algebraic solution for the intersection of two circles. Starting with the i^{th} and j^{th} circles in the form:

$$(t_x - x_{ci})^2 + (t_y - y_{ci})^2 = r_i^2$$
$$(t_x - x_{cj})^2 + (t_y - y_{cj})^2 = r_j^2$$

We can define the following geometric quantities. The square of the distance between their center points is:

$$L^2 = (x_{ci} - x_{cj})^2 + (y_{ci} - y_{cj})^2$$

The square of the difference in their radii is:

$$dR^2 = r_i^2 - r_j^2$$

The distance D of the perpendicular from the line joining the centerpoints to the points of intersection is:

$$D = [L^2(2(r_i^2 + r_j^2) - L^2) - (dR^2)^2]^{1/2}/(2L)$$

Then we can express the intersection points as:

$$P_1 = (X_1+X_2 , Y_1-Y_2)$$
$$P_2 = (X_1-X_2 , Y_1+Y_2)$$

where X_1, Y_1, X_2, and Y_2 are given by:

$$X_1 = [(x_{cj} - x_{ci})dR^2 + (x_{ci} + x_{cj})L^2]/(2L^2)$$
$$Y_1 = [(y_{cj} - y_{ci})dR^2 + (y_{ci} + y_{cj})L^2]/(2L^2)$$
$$X_2 = (y_{cj} - y_{ci})D/L$$
$$Y_2 = (x_{cj} - x_{ci})D/L$$

We must now apply a constraint based on θ to eliminate one of the two points. Using the same equation for θ as before we have:

$$d\theta_i = \text{ATAN2}(y_{2i}, x_{2i}) - \text{ATAN2}(y_{1i} - t_y, x_{1i} - t_x)$$
$$d\theta_j = \text{ATAN2}(y_{2j}, x_{2j}) - \text{ATAN2}(y_{1j} - t_y, x_{1j} - t_x)$$

These equations can be used to determine which of our two solutions is correct by yielding a consistent value of $d\theta$ when our two points P_1 and P_2 are substituted for (t_x,t_y). We define the *angle consistency error* as:

$$d\theta_{ij(con)} = \min[\, | d\theta_i - d\theta_j |_{Px1,Py1}, | d\theta_i - d\theta_j |_{Px2,Py2} \,]$$

The point pair P_1 or P_2 which yields $d\theta_{ij(con)}$ is our consistent solution. We also obtain a corresponding estimate for $d\theta$ based on the average of the two consistent values for $d\theta$, namely

$$d\Theta_k = (d\Theta_i + d\Theta_j)/2$$

Once we have determined the consistent set of points, we can take an ensemble average as follows:

$$t_x = P_{xavg} = (P_{x1} + P_{x2} + \ldots + P_{xN})/N$$
$$t_y = P_{yavg} = (P_{y1} + P_{y2} + \ldots + P_{yN})/N$$
$$d\Theta_{avg} = (d\Theta_1 + d\Theta_2 + \ldots + d\Theta_N)/N$$

$$\text{where } N = n(n-1)/2$$

The reciprocal of the *angle consistency error* can also be used as a weighting factor in calculating a weighted average for t_x, t_y, and $d\Theta$. This formulation is:

$$t_x = P_{xWavg} = (P_{x1}/d\Theta_{1(con)} + \ldots + P_{xN}/d\Theta_{N(con)})/(1/d\Theta_{1(con)} + \ldots + 1/d\Theta_{N(con)})$$
$$t_y = P_{yWavg} = (P_{y1}/d\Theta_{1(con)} + \ldots + P_{yN}/d\Theta_{N(con)})/(1/d\Theta_{1(con)} + \ldots + 1/d\Theta_{N(con)})$$
$$d\Theta_{Wavg} = (d\Theta_1/d\Theta_{1(con)} + \ldots + d\Theta_N/d\Theta_{N(con)})/(1/d\Theta_{1(con)} + \ldots + 1/d\Theta_{N(con)})$$